BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

IN THE CLAIMS:

1. (Currently amended) A method in a data processing system for creating instruction bundles, comprising:

receiving an instruction group having one or more instructions;

automatically determining a number of each possible type of instruction in the one or more instructions of the instruction group; and

dynamically creating one or more instruction bundles based on the number of each possible type of instruction in the one or more instructions of the instruction group.

2. (Original) The method of claim 1, wherein receiving an instruction group having one or more instructions includes receiving a stream of intermediate instructions organized into instruction groups.

3. (Original) The method of claim 1, further comprising gathering information about an architecture for use in creating instruction bundles.

4. (Original) The method of claim 3, wherein the information includes at least one of a number of each type of execution unit available in the architecture and a number of bundles that can be dispatched concurrently by the architecture.

5. (Original) The method of claim 2, wherein the steps of determining a number of each possible type of instruction and creating one or more instruction bundles is performed for each instruction group in the stream of intermediate instructions.

6. (Original) The method of claim 1, wherein creating one or more instruction bundles is performed in view of one or more of the following rules:

1) instructions of the same instruction type will preserve there original order;

2) branches will normally appear only in the final bundle of an instruction group;

3) for architectures where a number of M execution units is equal or less than a number of concurrent bundles, MM templates will only be used when there are three or fewer instructions remaining in the group;

4) instructions are taken in order of their flexibility in terms of where that instruction can be placed in the available bundle types; and

5) MBB and BBB templates are avoided when only a single B instruction remains.

7. (Original) The method of claim 1, wherein creating one or more instruction bundles is performed based on a most common instruction combination first.

8. (Original) The method of claim 7, wherein creating one or more instruction bundles is performed based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement second.

9. (Original) The method of claim 1, wherein creating one or more instruction bundles is performed based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement.

10. (Original) The method of claim 1, wherein determining a number of each possible type of instruction in the one or more instructions of the instruction group includes incrementing instruction counters based on the number of each possible type of instruction in the one or more instructions, and wherein creating one or more instruction bundles includes decrementing the instruction counters as instructions are added to instruction bundles.

11. (Original) The method of claim 7, wherein the most common instruction combination is where all instructions in the instruction group are of a memory instruction type, integer instruction type or integer arithmetic logic unit type.

12. (Original) The method of claim 1, wherein creating one or more instruction bundles includes ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription.

13. (Original) The method of claim 12, wherein ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription includes forming partial instruction bundles.

14. (Currently amended) An apparatus in a data processing system for creating instruction bundles, comprising:

means for receiving an instruction group having one or more instructions;

means for automatically determining a number of each possible type of instruction in the one or more instructions of the instruction group; and

means for dynamically creating one or more instruction bundles based on the number of each possible type of instruction in the one or more instructions of the instruction group.

15. (Original) The apparatus of claim 14, wherein the means for receiving an instruction group having one or more instructions includes means for receiving a stream of intermediate instructions organized into instruction groups.

16. (Original) The apparatus of claim 14, further comprising means for gathering information about an architecture for use in creating instruction bundles.

17. (Original) The apparatus of claim 16, wherein the information includes at least one of a number of each type of execution unit available in the architecture and a number of bundles that can be dispatched concurrently by the architecture.

18. (Original) The apparatus of claim 15, wherein the means for determining a number of each possible type of instruction and means for creating one or more

instruction bundles operate on each instruction group in the stream of intermediate instructions.

19. (Original) The apparatus of claim 14, wherein the means for creating one or more instruction bundles operates in view of one or more of the following rules:

1) instructions of the same instruction type will preserve there original order;

2) branches will normally appear only in the final bundle of an instruction group;

3) for architectures where a number of M execution units is equal or less than a number of concurrent bundles, MM templates will only be used when there are three or fewer instructions remaining in the group;

4) instructions are taken in order of their flexibility in terms of where that instruction can be placed in the available bundle types; and

5) MBB and BBB templates are avoided when only a single B instruction remains.

20. (Original) The apparatus of claim 14, wherein the means for creating one or more instruction bundles creates the one or more instruction bundles based on a most common instruction combination first.

21. (Original) The apparatus of claim 20, wherein the means for creating one or more instruction bundles creates the one or more instruction bundles based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement second.

22. (Original) The apparatus of claim 14, wherein the means for creating one or more instruction bundles creates the one or more instruction bundles based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement.

23. (Original) The apparatus of claim 14, wherein the means for determining a number of each possible type of instruction in the one or more instructions of the

instruction group includes means for incrementing instruction counters based on the number of each possible type of instruction in the one or more instructions, and wherein the means for creating one or more instruction bundles includes means for decrementing the instruction counters as instructions are added to instruction bundles.

24. (Original) The apparatus of claim 20, wherein the most common instruction combination is where all instructions in the instruction group are of a memory instruction type, integer instruction type or integer arithmetic logic unit type.

25. (Original) The apparatus of claim 14, wherein the means for creating one or more instruction bundles includes means for ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription.

26. (Original) The apparatus of claim 25, wherein the means for ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription includes means for forming partial instruction bundles.

27. (Currently amended) A computer program product in a computer readable medium for creating instruction bundles, comprising:

first instructions for receiving an instruction group having one or more instructions;

second instructions for automatically determining a number of each possible type of instruction in the one or more instructions of the instruction group; and

third instructions for dynamically creating one or more instruction bundles based on the number of each possible type of instruction in the one or more instructions of the instruction group.

28. (Original) The computer program product of claim 27, wherein the first instructions for receiving an instruction group having one or more instructions includes instructions for receiving a stream of intermediate instructions organized into instruction groups.

29. (Original) The computer program product of claim 27, further comprising fourth instructions for gathering information about an architecture for use in creating instruction bundles.

30. (Original) The computer program product of claim 29, wherein the information includes at least one of a number of each type of execution unit available in the architecture and a number of bundles that can be dispatched concurrently by the architecture.

31. (Original) The computer program product of claim 28, wherein the second instructions for determining a number of each possible type of instruction and third instructions for creating one or more instruction bundles are executed on each instruction group in the stream of intermediate instructions.

32. (Original) The computer program product of claim 27, wherein the third instructions for creating one or more instruction bundles are executed in view of one or more of the following rules:

1) instructions of the same instruction type will preserve there original order;

2) branches will normally appear only in the final bundle of an instruction group;

3) for architectures where a number of M execution units is equal or less than a number of concurrent bundles, MM templates will only be used when there are three or fewer instructions remaining in the group;

4) instructions are taken in order of their flexibility in terms of where that instruction can be placed in the available bundle types; and

5) MBB and BBB templates are avoided when only a single B instruction remains.

33. (Original) The computer program product of claim 27, wherein the third instructions for creating one or more instruction bundles creates the one or more instruction bundles based on a most common instruction combination first.

34. (Original) The computer program product of claim 33, wherein the third instructions for creating one or more instruction bundles creates the one or more instruction bundles based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement second.

35. (Original) The computer program product of claim 27, wherein the third instructions for creating one or more instruction bundles creates the one or more instruction bundles based on a most restrictive instruction type placement and proceeds to less restrictive instruction type placement.

36. (Original) The computer program product of claim 27, wherein the second instructions for determining a number of each possible type of instruction in the one or more instructions of the instruction group includes instructions for incrementing instruction counters based on the number of each possible type of instruction in the one or more instructions, and wherein the third instructions for creating one or more instruction bundles includes instructions for decrementing the instruction counters as instructions are added to instruction bundles.

37. (Original) The computer program product of claim 33, wherein the most common instruction combination is where all instructions in the instruction group are of a memory instruction type, integer instruction type or integer arithmetic logic unit type.

38. (Original) The computer program product of claim 27, wherein the third instructions for creating one or more instruction bundles includes instructions for ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription.

39. (Original) The computer program product of claim 38, wherein the instructions for ensuring that creating the one or more instruction bundles does not introduce hardware oversubscription includes instructions for forming partial instruction bundles.